EPA Proposal

Apprenticeship Tracker

By Abdul Sarwar

PwC

April 2025

Table of Contents

Problem Outline

This project addresses the lack of a centralised, accessible system for apprentices to track their development in Knowledge, Skills, and Behaviors (KSBs). Current solutions often lack personalisation and require external resources.

Derek Young highlights the benefits of electronic portfolios (ePortfolios) in education, stating:

"ePortfolios present the learner/practitioner with the capacity to act quickly on their reflections by offering the ability to adapt and restructure and by offering a range of technical tools to use both in reflective analysis and the presentation of skills development" (Young, 2008)

Young contrasts ePortfolios with traditional ones, stressing their ability for "constant" rather than "periodic" reflection (Young, 2008). A responsive digital tracker lets apprentices monitor progress, helps mentors align learning with standards, and facilitates timely feedback (Conveya, 2025).

Solution Outline

The Apprenticeship Tracker is a functional web-based platform aimed at supporting apprentices in their daily learning and development processes.

One of the primary components is the **Digital Diary**. This feature allows apprentices to log their daily activities, reflections, and feedback, thereby creating a detailed record of their development. The accumulated logs can also serve as a reference for mentors during evaluations, ensuring that feedback is grounded in documented experiences.

The platform's **Calendar** should aid apprentices in time management, allowing them to set alerts for key dates like assignment deadlines and project milestones. These reminders help apprentices manage their schedules effectively, reducing the risk of missed deadlines. Integrated with other platform features, it enhances organisational skills and boosts overall utility.

The **Benchmark Setting** feature should allow apprentices to establish milestones, such as mastering a skill or completing a project. These benchmarks are vital for tracking progress and assessing readiness for tasks like Reflective Statements. By achieving these goals, apprentices can better understand their progression.

User Requirements

Functional Requirements

The functional requirements of the software emphasise interactive features that are crucial for apprentices.

Apprentices should be able to enter and update logs related to their daily activities. These logs must be mapped against relevant Knowledge, Skills, and Behaviours (KSBs). The software should also allow apprentices to set and track SMART (Specific, Measurable, Achievable, Relevant, Time-bound) goals over time.

Apprentice User Story: "As an apprentice, I want to log my daily learning activities and map them to specific Knowledge, Skills, and Behaviours (KSBs) so that I can track my progress towards meeting End-Point Assessment (EPA) requirements efficiently." This solution offers a structured way to monitor learning and set SMART goals, a user-friendly and accessible interface, and secure data handling.

Non-Functional Requirements

Non-functional requirements emphasise usability, ensuring an intuitive, user-friendly interface suitable for varied digital skill levels (ATOLLO Project, 2024); reliable performance with fast loading across desktop and mobile devices (Mancuso, 2024); and accessibility, prioritising inclusive design through clear, readable fonts such as Inter to support users with disabilities or visual impairments (Burgstahler, 2020).

Security Requirements

Security requirements ensure that user data remains private and secure, similar to digital educational platforms that protect user information. Key security measures include that all personal data must be encrypted and stored securely. Role-based access control restricts sensitive records to authorised users, preventing unauthorised access, while audit trails log all system changes. Anonymisation and role-based access controls are integral parts of the platform to ensure user data is handled securely and responsibly.

Assumptions Made

Successful implementation relies on several assumptions, including User Willingness and Support Infrastructure. It's assumed that apprentices and coaches are motivated to engage regularly, and employers provide resources like Azure and GitHub. Comprehensive onboarding, regular coaching, and support teams are vital for enhancing proficiency. Fallback strategies and thorough documentation help overcome challenges and encourage software use.

Risk Analysis

Potential Issues

The handling of personal data introduces security risks. Compliance with the General Data Protection Regulation (GDPR) and adherence to stringent cybersecurity protocols mitigate these risks. These can be mitigated through the handling of personal data and introduced security risks.

Likelihood and Consequences

Integration issues present a moderate risk, potentially causing slower rollout and delays in full operational capacity. Low user engagement is a high-risk factor, significantly impacting the platform's ability to meet learning outcomes. Although data breaches are unlikely due to strong security measures, they have severe consequences, including legal liabilities and reputational damage.

Mitigation Strategies

To address these potential issues, several mitigation strategies can be implemented. To boost user engagement, incorporating elements of gamification and implementing notification systems can serve as effective strategies to encourage regular platform use. From a security perspective, adopting a security-by-design approach is crucial. This involves encrypting data, as well as applying role-based access controls.

Choice of Project Methodologies, Technologies, and Tools

Project Management Methodology

Agile project management, known for its flexibility and iterative development focus, is essential in educational tech projects. Agile methods foster "collaborative effort," allowing teams to deliver faster with higher quality (Agile Alliance, 2019). Azure DevOps will manage projects using Kanban boards to handle scope, plan short-term Agile tasks, and optimise time with sprints, benefiting from familiarity with the platform.

Design Methodologies

User-Centered Design (UCD) is implemented alongside Agile, involving real users in the design process. By iterating on wireframes and mockups with pilot users, developers can create products that are intuitive and functional, enhancing user engagement and retention (The Atollo Project, 2024). Figma is chosen for the design process due to my familiarity with its interface and features.

Implementation Technologies and Tools

The frontend employs React with Material-UI for responsive and scalable design. Material-UI, known for its large component library and adherence to Google's Material Design, facilitates the rapid creation of consistent and accessible UIs.It was chosen over appkit for its strong reputation, comprehensive documentation, and supportive community, with performance trade-offs being negligible for such a small project (GaelFerrand, 2021).

The backend uses Node.js with Express.js for real-time interactions and PostgreSQL for secure database management, offering strategic advantages. Node.js, with its non-blocking I/O model, is ideal for real-time applications, while Express.js provides flexibility and simplicity. PostgreSQL enhances security with robust access controls and encryption, ensuring data integrity. Both technologies are scalable. Using JavaScript across both client and server sides increases development flexibility.

To ensure application quality and reliability, Jest is used for automated testing, Postman for API testing, and GitHub Actions for continuous integration.

The code is stored on GitHub and deployed to Google Cloud Platform for deployment, utilising Firebase for authentication and role-based access control. Firebase's suite of cloud services facilitates secure data access and application deployment, ensuring scalability, resilience, and secure user data handling.

Testing Plan

Unit Testing

Using Jest, unit testing verifies individual components and functions, catching early errors in functionalities like diary entry submissions and benchmark creations.

Integration Testing

This phase ensures components work together seamlessly, verifying interactions such as diary inputs updating progress tracking and calendar alerts syncing with tasks.

User Acceptance Testing (UAT)

Apprentices and coaches test key user journeys, providing feedback on log submissions, goal setting.

Continuous Testing

Automated pipelines through GitHub Actions test code changes, ensuring consistent, reliable delivery by detecting bugs early and preventing regression issues.

Below is an outline of a testing table. A more comprehensive one will need to be drawn up once more of the system has been planned.

Table 1: Test Phase Summary for Apprenticeship Tracker Application

Test Phase	Test Case ID	Component /Scenario	Test Description	Expected Outcome	Actual Outcome
Unit Testing	UT-001	Diary Entry Submission	Verify the ability to submit and save daily diary entries.	Entry is saved successfully and displayed in the diary log.	[Pass /Fail]
	UT-002	Benchmark Creation	Verify ability to set and save benchmarks.	Benchmark is created and displayed on a progression tracker.	[Pass /Fail]
	UT-003	SMART Goal Tracking	Validate the setting and updating of SMART goals.	Goals should be easily set and updated with real-time tracking.	[Pass /Fail]
Integration Testing	IT-001	Diary, Progress Tracking	Ensure diary input updates progress tracking.	Diary entry accurately reflects the progress dashboard.	[Pass /Fail]
	IT-002	Calendar, Task Sync	Verify synchronisation between calendar alerts and task deadlines.	Alerts trigger and display at correct times.	[Pass /Fail]
User Acceptance Testing (UAT)	UAT-001	Apprentice Log Submission	Apprentice logs daily activities and maps KSBs.	Entries are saved and mapped correctly.	[Pass /Fail]
	UAT-002	Coach Dashboard Review	Dashboard works	Dashboard displays accurate apprentice data.	[Pass /Fail]

Project Plan

The project will be executed in six key phases over 12 weeks, following an Agile delivery schedule. Each phase includes defined milestones.

Initiation & Planning: Define project scope, gather comprehensive user requirements, develop user personas, and draft technical specifications.

Design Phase: Create low- and high-fidelity wireframes and finalise UI components. Integrate features like the user dashboards, and goal tracking

- **Milestone**: The final UI design and Feature-complete MVP. The final UI design MVP phase involves developing low and high-fidelity wireframes. This iterative process enhances the user interface, adding features like dashboards and goal-tracking tools. The goal is to finalise an intuitive and appealing design, ready for testing and evaluation.

Development Phase: Implement core features including the digital diary, KSB mapping, and calendar integration.

- **Milestone**: Completion of core feature set. This includes building features like the digital diary for documenting activities, Knowledge, Skills, and Behaviours (KSB) mapping, and calendar integration for effective time management. The successful completion of this phase is marked by having all initial features developed and operational, ready for thorough testing.

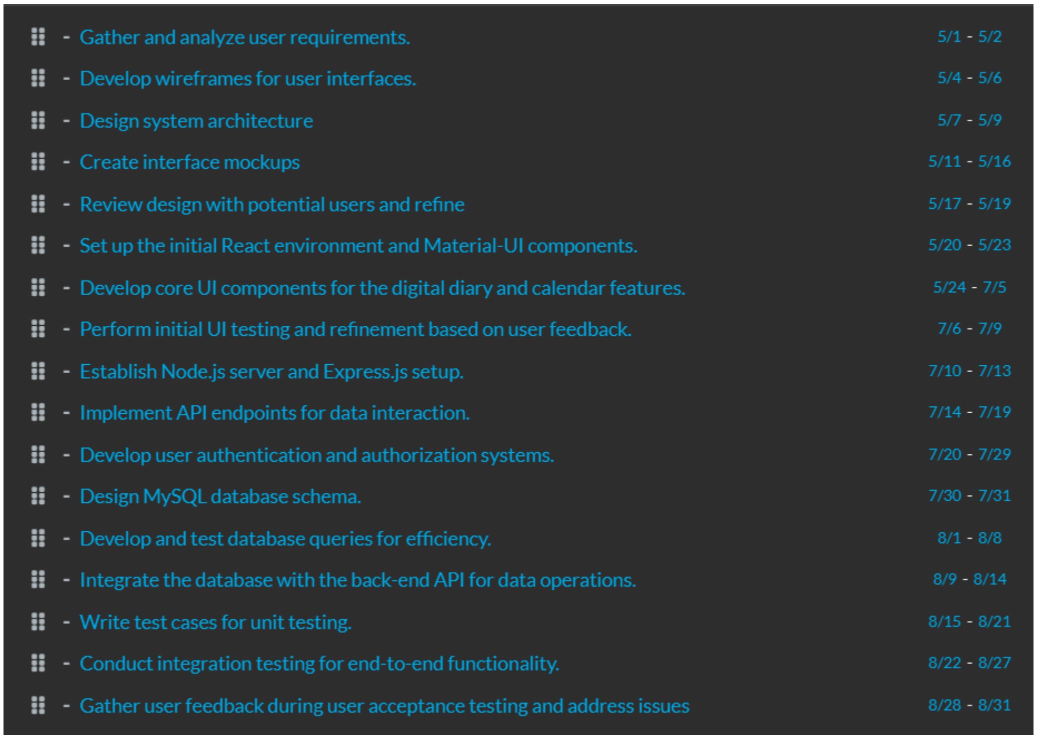
Testing Phase: Perform unit, integration, and user acceptance tests to fix bugs and refine the UI.

- **Milestone**: Passed acceptance tests. The testing phase ensures application quality through unit testing for components, integration testing for module interaction, and user acceptance testing (UAT) for end-user feedback. This confirms the software meets user requirements and is free of bugs.

Deployment & Feedback: Deploy the MVP, onboard test users, and gather feedback for future improvements.

- **Milestone**: Successful deployment and feedback collection. The MVP of the Apprenticeship Tracker is deployed to production, involving test user onboarding and feedback collection to pinpoint improvements. Success is measured by stable operation and positive feedback, paving the way for future iterations and feature expansions.

Below is my GANTT chart. The lines connecting each item represents if the task is dependent on the other.



- Gather and analyze user requirements.	5/1 - 5/2
- Develop wireframes for user interfaces.	5/4 - 5/6
- Design system architecture	5/7 - 5/9
- Create interface mockups	5/11 - 5/16
- Review design with potential users and refine	5/17 - 5/19
- Set up the initial React environment and Material-UI components.	5/20 - 5/23
- Develop core UI components for the digital diary and calendar features.	5/24 - 7/5
- Perform initial UI testing and refinement based on user feedback.	7/6 - 7/9
- Establish Node.js server and Express.js setup.	7/10 - 7/13
- Implement API endpoints for data interaction.	7/14 - 7/19
- Develop user authentication and authorization systems.	7/20 - 7/29
- Design MySQL database schema.	7/30 - 7/31
- Develop and test database queries for efficiency.	8/1 - 8/8
- Integrate the database with the back-end API for data operations.	8/9 - 8/14
- Write test cases for unit testing.	8/15 - 8/21
- Conduct integration testing for end-to-end functionality.	8/22 - 8/27
- Gather user feedback during user acceptance testing and address issues	8/28 - 8/31

Figure 1 (Above): List of tasks in GANTT chart

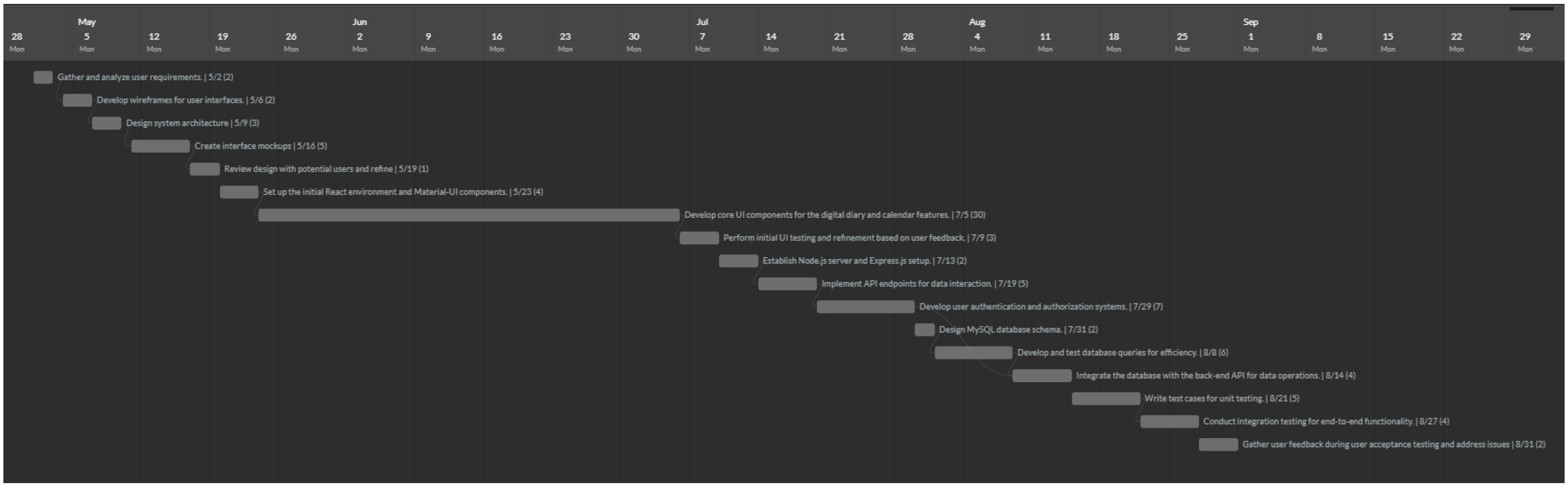


Figure 2 (Above): GANTT chart of proposed length of each task and representation of dependencies

References

Mancuso, O. (2024). What Makes Apprentix the Best Apprenticeship Management Software and Apprenticeship Platform? [online] Apprentix.io. Available at: https://www.apprentix.io/post/what-makes-apprentix-best-apprenticeship-management-software-apprenticeship-platform [Accessed 28 Mar. 2025]

The Atollo Project (2024). User-centred Design In Educational Tools. [online] Available at: https://atolloproject.eu/the-role-of-user-centred-design-in-educational-tools/ [Accessed 28 Mar. 2025]

Bcs.org (2019). Deliver the digital apprenticeship standards. [online] Available at: https://www.bcs.org/deliver-and-teach-qualifications/training-providers-and-adult-education-centres/deliver-digital-it-apprenticeships/deliver-our-apprenticeship-standards/ [Accessed 28 Mar. 2025]

Burgstahler, S. (2020). Equal Access: Universal Design of Instruction. University of Washington. [online] Available at: https://www.washington.edu/doit/equal-access-universal-design-instruction [Accessed 28 Mar. 2025]

Doppelt, Y., Mehalik, M.M., Schunn, C.D., Silk, E.M. and Krysinski, D. (2008). Engagement and Achievements: A Case Study of Design-Based Learning in a Science Context. Journal of Technology Education, 19(2), pp.22–39.

EDlounge. (n.d.). Apprenticeship ePortfolio and Assessment Platform. [online] Available at: https://scholar.lib.vt.edu/ejournals/JTE/v19n2/pdf/doppelt.pdf [Accessed 28 Mar. 2025]

GoSprout (2025). Apprenticeship Management Platform | GoSprout. [online] Available at: https://gosprout.app/apprenticeship-management-platform/ [Accessed 28 Mar. 2025]

Institute for Apprenticeships and Technical Education (2024). Developing an occupational standard. [online] Institute for Apprenticeships and Technical Education. Available at: https://www.instituteforapprenticeships.org/developing-new-apprenticeships/developing-occupational-standards/. [Accessed 28 Mar. 2025]
Oviatt, S. (2013). The Design of Future Educational Interfaces. New York: Routledge

Queen Mary University of London. (n.d.). Study Time - Human Resources. [online] Available at: https://www.qmul.ac.uk/human-resources/organisational-professional-development/qualification-programmes/managing-an-apprentice/study-time/ [Accessed 28 Mar. 2025]

Young, D.D. (2008). e-Portfolios in Education: a cause for reflection ‘eLearning and Learning Environments for the Future’. [online] Available at: https://dera.ioe.ac.uk/id/eprint/13076/6/6213.pdf [Accessed 28 Mar. 2025].

Conveya.co. (2025). Why you need to start using technological solutions to manage your apprenticeship programmes. [online] Available at: https://www.conveya.co/blog/why-you-need-to-start-using-technological-solutions-to-manage-your-apprenticeship-programmes [Accessed 28 Mar. 2025]

Material-UI (2025). Material-UI: A popular React UI framework. [online] Available at: https://mui.com/material-ui/getting-started/overview/ [Accessed 28 Mar. 2025]

Ansar (2023). Agile Project Management for Education: A Comprehensive Guide. [online] Teamhub. Available at: https://teamhub.com/blog/agile-project-management-for-education-a-comprehensive-guide/. [Accessed 28 Mar. 2025]

Agile Alliance (2019). What is Agile Software Development? [online] Agile Alliance. Available at: https://www.agilealliance.org/agile101/ [Accessed 28 Mar. 2025]

GaelFerrand (2021). Why you should NOT use Material-UI. [online] DEV Community. Available at: https://dev.to/gaelferrand/why-you-should-not-use-material-ui-21nn [Accessed 4 Apr. 2025].